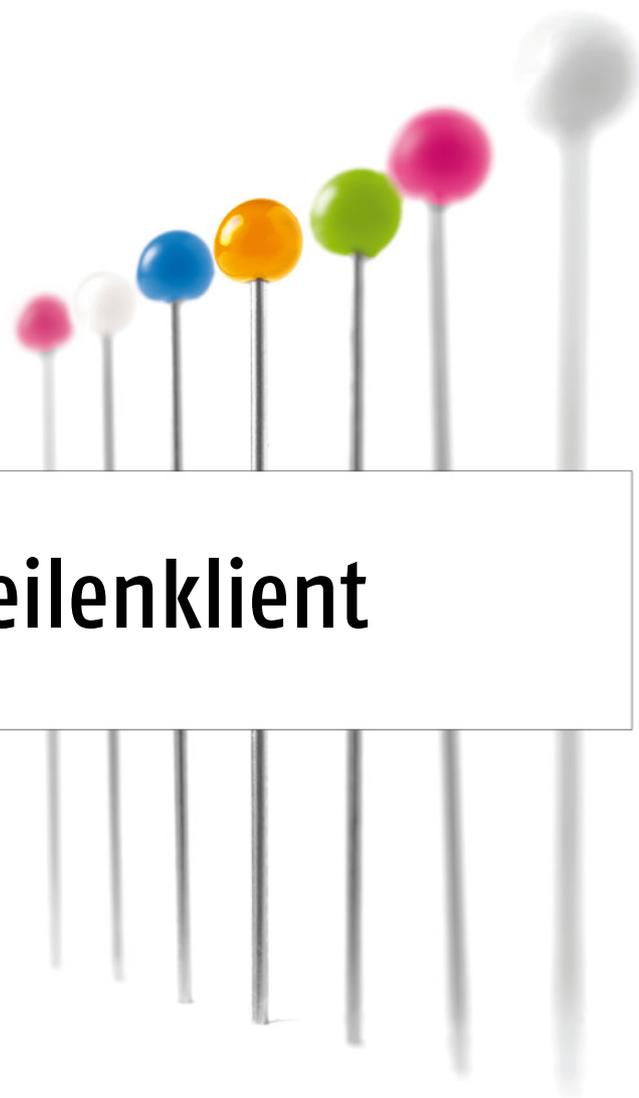


NOKIS 2.0



CSW-Kommandozeilenklient

Inhaltsverzeichnis

1	Impressum und Kontakt	5
2	Allgemeines	7
3	Installation	7
4	Verwendung.....	7
4.1	Aufruf und Parameter.....	7
4.2	CSW-Implementierungen testen.....	8
4.2.1	Die Hilfe aufrufen	8
4.2.2	XML-Dokument für die Anfrage an den Server angeben	8
4.2.3	Dateiname für die Speicherung der Antwort des Servers angeben	8
4.2.4	URL für das GetCapabilities-Dokument angeben	8
4.2.5	Optional: Benutzernamen für die Anfrage an den Server angeben	9
4.2.6	Optional: Passwort für die Anfrage an den Server angeben	9
4.2.7	Optional: Anfrage und Antwort validieren	9
4.2.8	Optional: Fehlerausgabe steuern	9
4.3	Erweitertes Logging	9
5	Anhang: Beispiele	10
5.1.1	Hilfeseite aufrufen.....	10
5.1.2	CSW-Server anfragen	10
5.1.3	CSW-Server mit Authentifizierung anfragen.....	10

1 Impressum und Kontakt

Copyright © 2010 disy Informationssysteme GmbH
Alle Rechte vorbehalten

Die Urheberrechte dieses Dokuments liegen bei der disy Informationssysteme GmbH.

Jegliche Kopien dieses Dokuments oder Auszüge davon müssen diese Copyright-Mitteilung enthalten, neben allen sonstigen Mitteilungen zum Schutz von Urheberrechten, die in den Materialien enthalten sind.

In dieser Publikation werden Warenzeichen, eingetragene Warenzeichen, Handelsnamen und Gebrauchsnamen verwendet. Auch wenn diese nicht als solche gekennzeichnet sind, gelten die entsprechenden Schutzbestimmungen.

Aufgrund ständiger Weiterentwicklung des Produkts behalten wir uns Änderungen vor. Wir garantieren nicht, dass dieses Dokument fehlerfrei ist. Über Probleme mit diesem Dokument informieren Sie uns bitte.

disy Informationssysteme GmbH
Erbprinzenstraße 4–12
76133 Karlsruhe

Tel.: +49 721 16006-000
Fax.: +49 721 16006-05

E-Mail: preludio@disy.net
<http://www.disy.net>

2 Allgemeines

Bei dem CSW-Klienten handelt es sich um ein Werkzeug, mit dem beliebige CSW 2.0.2-Implementierungen getestet werden können. Der Aufruf erfolgt von der Kommandozeile aus und erlaubt die Festlegung der GetCapabilities-URL, des Dateinamens der Datei mit dem Request-XML und des Dateinamens für die Speicherung der Antwort vom Server. Die Zielgruppe für die Verwendung des Werkzeugs sind Experten, die mit einem einfachen Werkzeug in die Lage versetzt werden sollen, beliebige CSW-Implementierungen mit vordefinierten Anfragen zu testen.

3 Installation

disy liefert den CSW-Klienten in Form eines Installers für Windows und Linux aus. Ein zu dem Klienten passendes JRE wird mitgeliefert. Während des Installationsvorgangs kann der Installationsort angepasst werden.

4 Verwendung

4.1 Aufruf und Parameter

Aufruf

Die Syntax des Aufrufs ist einfach gehalten. Es gibt keine grafische Benutzerschnittstelle zu dem Werkzeug, d- h. der Aufruf des CSW-Klienten erfolgt ausschließlich auf der Kommandozeile mit `csw202` (und der Angabe von zusätzlichen Parametern).

Beispiel: `csw202 -h`

Mit diesem Aufruf wird die Hilfe angezeigt.

Achtung Linux-Benutzer: je nachdem ob das aktuelle Verzeichnis (".") im Pfad enthalten ist oder nicht, muss der Programmaufruf explizit erfolgen: `./csw202`.

Die Parameter

Die folgende Tabelle gibt einer Übersicht der möglichen Parameter zum Aufruf des CSW-Klienten.

Syntax – kurz	Syntax – lang	Kurzbeschreibung
<code>-h</code>	<code>--help</code>	Print this message
<code>-i</code>	<code>--input <arg></code>	Input resource location
<code>-o</code>	<code>--output <arg></code>	Output resource location
<code>-pw</code>	<code>--password <arg></code>	Password
<code>-u</code>	<code>--get-capabilities-url <arg></code>	URL of the CSW GetCapabilities service
<code>-un</code>	<code>--username <arg></code>	Username

4.2 CSW-Implementierungen testen

4.2.1 Die Hilfe aufrufen

Syntax – kurz: `-h`
Syntax – lang: `--help`

Mit der **alleinigen** Angabe dieses Parameters wird die Kurzhilfe angezeigt. Die Kurzhilfe beschreibt kurz die möglichen Parameter.

4.2.2 XML-Dokument für die Anfrage an den Server angeben

Syntax – kurz: `-i <arg>`
Syntax – lang: `--input <arg>`

Mit diesem Parameter wird die URI der zu verwendenden Eingabedatei bestimmt. Dabei kann es sich um eine beliebige URI (lokale Datei, Datei im Web etc.) handeln. Im Normalfall wird hier eine lokale XML-Datei referenziert (Beispiel: [eingabe.xml](#)):

```
-i eingabe.xml  
  
--input eingabe.xml
```

Bei der Eingabedatei muss es sich um eine Datei handeln, in der ein valides Anfragedokument enthalten ist.

4.2.3 Dateiname für die Speicherung der Antwort des Servers angeben

Syntax – kurz: `-o <arg>`
Syntax – lang: `--output <arg>`

Mit diesem Parameter wird der Dateiname für die Ausgabedatei bestimmt. Dabei muss es sich um eine lokale Datei handeln und der ausführende Benutzer muss Schreibrechte für das entsprechende Verzeichnis besitzen (Beispiel: [ausgabe.xml](#)):

```
-o ausgabe.xml  
  
--output ausgabe.xml
```

4.2.4 URL für das GetCapabilities-Dokument angeben

Syntax – kurz: `-u <arg>`
Syntax – lang: `--get-capabilities-url <arg>`

Mit diesem Parameter wird die URL des GetCapabilities-Dokuments der abzufragenden CSW-Instanz festgelegt. (Beispiel:

```
http://www.meinhost.de/csw?request=GetCapabilities):
```

```
-u http://www.meinhost.de/csw?request=GetCapabilities  
  
-- get-capabilities-url  
http://www.meinhost.de/csw?request=GetCapabilities
```

Aus dem GetCapabilities-Dokument entnimmt der CSW-Klient dann die weiteren benötigten Informationen.

Achtung Linux-Benutzer: die URL des Kataloges kann Sonderzeichen enthalten, die von der Shell interpretiert werden (z.B. "\$"). In diesem Fall muss die URL in einfache Anführungszeichen (') gesetzt werden um eine Substitution zu unterbinden.

4.2.5 Optional: Benutzernamen für die Anfrage an den Server angeben

Syntax – kurz: `-un <arg>`
Syntax – lang: `--username <arg>`

Mit diesem optionalen Parameter wird der Benutzername spezifiziert. Dies ist in den Fällen notwendig, in denen die Ausführung der Anfrage eine Authentifizierung erfordert (Beispiel: jochen.mustermann):

```
-un jochen.mustermann  
--username jochen.mustermann
```

4.2.6 Optional: Passwort für die Anfrage an den Server angeben

Syntax – kurz: `-pw <arg>`
Syntax – lang: `--password <arg>`

Mit diesem optionalen Parameter wird das Passwort spezifiziert. Dies ist in den Fällen notwendig, in denen die Ausführung der Anfrage eine Authentifizierung erfordert (Beispiel: supergeheim):

```
-pw supergeheim  
--password supergeheim
```

4.2.7 Optional: Anfrage und Antwort validieren

Syntax – kurz: `-v`
Syntax – lang: `--validate`

Mit diesem optionalen Parameter wird die Validierung der Ein- und Ausgabedokumente eingeschaltet. Bei gesetztem Parameter werden die Dokumente gegen die offiziellen OGC-Schemata validiert und es werden Validierungsinformationen ausgegeben.

4.2.8 Optional: Fehlerausgabe steuern

Syntax – kurz: `-e`
Syntax – lang: `--errors`

Mit diesem optionalen Parameter wird die Ausgabe von Fehlermeldungen gesteuert. Wird der Parameter angegeben, so werden Fehlermeldungen des CSW-Klienten auf der Konsole ausgegeben.

4.3 Erweitertes Logging

Der CSW-Klient enthält umfangreiche Möglichkeiten, die Kommunikation zwischen dem Klienten und dem Server anhand von Logausgaben zu überwachen. Zur Konfiguration dieser Option befindet sich im Verzeichnis `classes` die Datei `log4j.properties`. In dieser Datei gibt es im Wesentlichen drei Zeilen, die den Umfang des Loggings steuern:

```
1: # HttpClient
2: # Uncomment to log HTTP request and response information
3: # log4j.logger.org.apache.commons.httpclient.HttpMethod=TRACE
4: # Uncomment to log low-level HTTP request and response ...
5: # log4j.logger.httpclient.wire.header=DEBUG
6: # log4j.logger.httpclient.wire.content=DEBUG
```

Zeile 3: Logausgaben abschalten

Die Zeile 3 steuert den Umfang der Informationen, die über die HTTP-Anfragen und Antworten ausgegeben werden. Wenn das Kommentarzeichen (#) am Anfang der Zeile entfernt wird, so werden keine diesbezüglichen Informationen ausgegeben.

Zeile 4 und 5: Weiterführende Informationen anzeigen

Die Zeilen 4 und 5 geben noch detailliertere Informationen über die HTTP-Kommunikation aus. Diese Option sollte nur verwendet werden, wenn mit der ersten Option noch nicht genug Informationen gesammelt werden konnten. Auch hier kann diese Option aktiviert werden, indem das Kommentarzeichen (#) am Anfang der Zeile entfernt wird.

Beispielkonfiguration aktivieren

Neben der standardmäßig aktivierten `log4j.properties` ist in der Auslieferung auch eine Beispielkonfiguration zur Ausgabe der Logmeldungen in eine Datei `log4j.properties.file` enthalten. Um diese Konfiguration zu aktivieren muss die vorhandene Datei umbenannt werden (z. B. `log4j.properties.konsole`) und die Datei `log4j.properties.file` muss umbenannt werden in `log4j.properties`.

5 Anhang: Beispiele

5.1.1 Hilfeseite aufrufen

Aufruf mit Parameter: `csw202 -h`

5.1.2 CSW-Server anfragen

Aufruf mit Parametern:

```
csw202 -u http://www.meinhost.de/csw?request=GetCapabilities -i
eingabe.xml -o ausgabe.xml
```

Mit diesem Befehl wird der in der Datei `eingabe.xml` abgelegte XML-Request an den Server `http://www.meinhost.de` geschickt. Die Antwort wird in der Datei `ausgabe.xml` abgelegt.

5.1.3 CSW-Server mit Authentifizierung anfragen

Aufruf mit Parametern:

```
csw202 -u http://www.meinhost.de/csw?request=GetCapabilities -i
eingabe.xml -o ausgabe.xml -un jochen.mustermann -pw supergeheim
```

Dieses Beispiel ist nahezu identisch mit dem vorhergehenden. In diesem Beispiel wird dem Request aber noch ein Benutzername und ein Passwort mitgegeben, mit denen sich der CSW-Klient beim CSW-Server authentifiziert. Die unterstützte Authentifizierungsmethode ist Basic HTTP Authentication.

Hinweis: Der Benutzername und Passwort werden unverschlüsselt an den Server übermittelt. Falls die Kommunikation nicht über verschlüsseltes HTTP (also HTTPS) erfolgt, dann können der Benutzername und das Passwort von unbefugten Dritten, die Zugriff auf den Netzwerkverkehr haben, ausgelesen werden.